09/936203

10 Rec'd PCT/PTO 10 SEP 2001 PCT

S&H Form. FORM PTO-1390 (2/01)

OIPE JC181 SEP 1 0 2001 PATENT & TRADEMARK OFFICE

U S DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE **TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371**	ATTORNEY'S DOCKET NUMBER 1454.1090

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/DE/00/00738	March 9, 2000	March 9, 1999

TITLE OF INVENTION
SYSTEM AND METHOD FOR IDENTIFYING OBJECTS IN DISTRIBUTED HIERARCHICAL SYSTEMS, ESPECIALLY AUTOMATION SYSTEMS

APPLICANT(S) FOR DO/EO/US
Norbert BECKER et al.

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. [X] This is a FIRST submission of items concerning a filing under 35 U.S.C. 371.
2. [X] This is an express request to immediately begin national examination procedures (35 U.S.C. 371(f)).
3. [X] The US has been elected by the expiration of 19 months from the priority date (PCT Article 31).
4. [X] A copy of the International Application as filed (35 U.S.C. 371(c)(2))
   a. [X] is transmitted herewith (required only if not transmitted by the International Bureau).
   b. [ ] has been transmitted by the International Bureau.
   c. [ ] is not required, as the application was filed in the United States Receiving Office (RO/US).
5. [X] A translation of the International Application into English (35 U.S.C. 371(c)(2)).
6. [ ] Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
   a. [ ] are transmitted herewith (required only if not transmitted by the International Bureau).
   b. [ ] have been transmitted by the International Bureau.
   c. [ ] is not required, as the application was filed in the United States Receiving Office (RO/US)
7. [ ] A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
8. [X] An oath or declaration of the inventor (35 U.S.C. 371(c)(4)).
9. [ ] A translation of the Annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 10-15 below concern document(s) or information included:

10. [X] An Information Disclosure Statement Under 37 CFR 1.97 and 1.98.
11. [ ] An assignment document for recording.
    Please mail the recorded assignment document to:
    a. [ ] the person whose signature, name & address appears at the bottom of this document.
    b. [ ] the following:
12. [X] A preliminary amendment.
13. [X] A substitute specification
14. [ ] A change of power of attorney and/or address letter.
15. [ ] Other items or information:

09/936203
518 Rec'd PCT/PTO 10 SEP 2001

[X] The U.S. National Fee (35 U.S.C. 371(c)(1)) and other fees as follows:

CLAIMS	(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) CALCULATIONS
	TOTAL CLAIMS	18 -20=	*	x $ 18.00	0.00
	INDEPENDENT CLAIMS	2 -3=	*	x $ 80.00	0.00
	MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+$270.00	0.00
	BASIC NATIONAL FEE (37 CFR 1.492(a)(1)-(4): [ ] Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO ....$1,000 [ ] International preliminary examination fee (37 C.F.R. 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO.. ....$ 860 [ ] International preliminary examination fee (37 C.F.R. 1.482) not paid to USPTO but international search fee (37 C.F.R. 1.445(a)(2) paid to USPTO..................$ 710 [ ] International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provision of PCT Article 33(1)-(4).............$ 690 [ ] International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(2) to (4) ....$ 100				860.00
	Surcharge of $130 for furnishing the National fee or oath or declaration later than [ ] 20 [ ] 30 mos. from the earliest claimed priority date (37 CFR 1.482(e)).				0.00
			TOTAL OF ABOVE CALCULATIONS		860.00
	Reduction by 1/2 for filing by small entity, if applicable. Affidavit must be filed also. (Note 37 CFR 1.9, 1.27, 1.28.)				
			SUBTOTAL		860.00
	Processing fee of $130 for furnishing the English Translation later than [ ] 20 [ ] 30 mos. from the earliest claimed priority date (37 CFR 1.482(f)).				
			TOTAL NATIONAL FEE		860.00
	Fee for recording the enclosed assignment (37 CFR 1.21(h)).				+
			TOTAL FEES ENCLOSED		860.00

a. [X] A check in the amount of $ 860.00 to cover the above fees is enclosed.
b. [ ] Please charge my Deposit Account No. 19-3935 in the Amount of $ to cover the above fees. A duplicate copy of this sheet is enclosed.
c. [X] The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 19-3935. A duplicate copy of this sheet is enclosed.

21171
PATENT TRADEMARK OFFICE

9/10/01
DATE

Richard A. Gollhofer
NAME Richard A. Gollhofer
REGISTRATION NO. 31,106

09/936203
518 Rec'd PCT/PTO 10 SEP 2001

Docket No. 1454.1090/RAG

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Norbert BECKER et al.

Application No.: Group Art Unit:

Filed: (Concurrently) Examiner:

For: SYSTEM AND METHOD FOR IDENTIFYING OBJECTS IN DISTRIBUTED HIERARCHICAL SYSTEMS, ESPECIALLY AUTOMATION SYSTEMS (as amended)

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Before examination of the above-identified application, please amend the application as follows:

IN THE TITLE:

Please change "ESPECIALLY" to --IN PARTICULAR IN--.

IN THE SPECIFICATION:

Please REPLACE the pending specification with the Substitute Specification attached hereto.

IN THE ABSTRACT:

Please REPLACE the originally filed Abstract with the enclosed Substitute Abstract.

IN THE CLAIMS:

Please CANCEL claims 1-2 without prejudice or disclaimer of any of the subject matter claimed therein and ADD new claims in accordance with the following:

3. (NEW) A system for identifying objects in a distributed hierarchical system, comprising:

a storage unit to store identifiers characterizing the objects and contexts for managing the identifiers, the contexts forming a plurality of indirection stages and, within a context, names and identifiers for all contained objects are known and unique.

4. (NEW) The system as claimed in claim 3, the storage unit further storing container objects and context information for each container object, with the container objects containing other objects and the context information being hierarchically structured.

5. (NEW) The system as claimed in claim 4, the storage unit further storing additional contexts which are not hierarchically related that may be unidirectionally associated with any desired context to access information thereof.

6. (NEW) The system as claimed in claim 4, wherein the objects have associated local object identifications stored in a lowest possible container object containing the objects, and the objects are able to be identified using a chain of object identifications.

7. (NEW) The system as claimed in claim 3, wherein the contexts include identification contexts for managing context identifications, the context identifications being associated with objects in the identification context and being valid and unique within the identification context.

8. (NEW) The system as claimed in claim 3, wherein connections between the objects are in the form of monikers.

9. (NEW) The system as claimed in claim 3, wherein a document is provided as the smallest environment for a context, with context information associated with the document being used for objects embedded in the document, and the document having a name allocated thereto.

10. (NEW) The system as claimed in claim 3, wherein the contexts are provided for managing the identifiers of the objects for object operations including at least one of moving, copying and renaming, without global, central management functions being provided.

11. (NEW) The method as claimed in claim 3, wherein the distributed hierarchical system is an automation system.

12. (NEW) A method for identifying objects in a distributed hierarchical system in which objects are characterized with identifiers, and contexts manage the identifiers, comprising:

storing the contexts to form a plurality of indirection stages, with all names and identifiers for all contained objects being known and unique within a context.

13. (NEW) The method as claimed in claim 12, further comprising:

storing container objects that contain other objects, and hierarchically structured context information for each container object.

14. (NEW) The method as claimed in claim 13, further comprising:

unidirectionally associating additional contexts, which are not hierarchically related, with any desired context to access information thereof.

15. (NEW) The method as claimed in claim 13, further comprising:

storing in a lowest possible container object containing the objects, local object identifications to identify the objects using a chain of object identifications.

16. (NEW) The method as claimed in claim 12, wherein at least some of the contexts manage context identifications associated with the objects in a corresponding context and being valid and unique within the corresponding context.

17. (NEW) The method as claimed in claim 12, further comprising:

storing connections between the objects are in the form of monikers.

18. (NEW) The method as claimed in claim 12, further comprising:

storing at least one document as the smallest environment for a context, with context information associated with the document being used for objects embedded in the document, and the document having a name allocated thereto.

19. (NEW) The method as claimed in claim 12, wherein the contexts are provided for managing the identifiers of the objects for object operations including at least one of moving, copying and renaming, without global, central management functions being provided.

20. (NEW) The method as claimed in claim 12, wherein the distributed hierarchical system is an automation system.

REMARKS

This Preliminary Amendment is submitted to improve the form of the English translation as filed. It is respectfully requested that this Preliminary Amendment be entered in the above-referenced application.

In accordance with the foregoing, claims 1-2 have been canceled and claims 3-20 have been added. Thus, claims 3-20 are pending and are under consideration.

A substitute specification is also being filed herewith. The substitute specification is accompanied by a marked-up substitute specification. No new matter has been added.

If there are any questions regarding these matters, such questions can be addressed by telephone to the undersigned. Otherwise, an early action on the merits is respectfully solicited.

If any further fees are required in connection with the filing of this Preliminary Amendment, please charge same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

By: Richard A. Gollhofer
Richard A. Gollhofer
Registration No. 31,106

700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001
(202) 434-1500

Date: 9/10/01

09/936203
518 Rec'd PCT/PTO 10 SEP 2001

GR 99 P 3135

2/prts

Description

System and method for object identification in distributed hierarchical systems, in particular in automation systems

The invention relates to a system and method for object identification in distributed hierarchical systems, in particular in automation systems.

Such a system and method are used particularly in the field of automation technology.

The invention is based on the object of ensuring object identification for operations such as moving, copying, renaming, etc.

This object is achieved by a system having the features specified in claim 1 and by a method having the features specified in claim 2.

The invention is based on the insight that previous solutions are not very robust and/or require a great deal of modification effort. There are two basic identification mechanisms which are used (and can also be combined with one another). One method is based on object identification by allocating a globally unique identifier for each object. This global identifier is used to ensure that an object can be found again irrespective of its present whereabouts. This method has the following drawbacks:

- **Central management**: The method requires central management structures such as management of the object identifiers and conversion tables for the object identifiers to the objects.
- **Poor support for distributed work**: The need for central management complicates

the splitting of object sets, separate processing and subsequent merging thereof (headword: branch and merge).

In the second method, an object is identified by its relative position with respect to another. This then also stipulates how the object can be found. In contrast to the first method, an object does not have a unique identifier, but instead the identifier [lacuna] dependent on the respective starting object referencing the other one. This means that no central management information is required. However, the following drawbacks result:

- **Low robustness**: Using the relative position for identification means that the identifier (or the identifiers) becomes invalid when the object is moved, and the object is no longer available (broken link).
- **Great deal of modification effort**: When the identifiers for an object have become invalid, they need to be corrected using a type of correction operation.

With the inventive solution, contexts are introduced for forming a plurality of indirection stages for managing the identifiers. This provides efficient methods for repairing "broken links" without introducing global, central management functions.

- **No central management**: Management is through the context hierarchy. This means that each context contains all the necessary information.
- **Support for distributed work**: The context hierarchy can be broken down and reassembled as desired. This means that projects can be branched and merged without difficulty.
- **Little modification effort**: The context hierarchy makes it immediately clear where changes to

identifiers are to be reconstructed. The changes also need to be made only on the context objects in question.

The invention is described and explained in more detail below using the exemplary embodiments shown in the figures, in which:

FIGURE 1 shows a block diagram to identify the facts of the matter: a client sees names, an object model works with IDs,

FIGURE 2 shows a schematic illustration for the allocation and assignment of object identifications as object IDs, and

FIGURE 3 shows a schematic illustration for the moving of an object denoted by "ES-Auto1".

The method is described below within the context of the OVA engineering object model (OVA= open distributed automation). It can also be used for other object models, however. For a better understanding of the relationships, a brief description of the context of the invention will be given below:

For each object, there is an environment in which it is known. In the case of OVA, this environment is modeled by the context. Within a context, names and identifiers for all contained objects are known and unique. The context is generally determined by the point of entry chosen by a user for processing his automation solution. Context information is available on any container object (i.e. an object containing other objects), such as H-container, chart or master. The smallest environment for a context is a document, however. For the case of embedded objects, the context information of the surrounding document is used. However, this also means that context information can be structured hierarchically. In this case, contexts at a lower point are always part of the higher context in the hierarchy as well. In addition, further contexts

which are not hierarchically related can be associated with any desired context. This context can then access the information of the associated context. This association is unidirectional, however. In the case of associated contexts, the (hierarchically) contained contexts are also associated automatically.
The context information determines which objects have been entered in the directory. The content of the document automatically belongs to the same context; this also applies, in particular, to linked objects or objects which have been added using a rule ("all objects in this directory", etc.).
The context is also the manager of the context IDs. These are described later.

Object identification

Since OVA uses a standardized method to access the data storage, it is necessary to structure the objects so that they can be safely identified. The reason for this is that some of the data are structure information, for example which ES-Auto is located in which chart, or which auto is connected to which. This structure information is subject to rules which are taken into account by the implementation of the data model. In the case of the current realization using the IStorage interface as the interface for data storage, it is always possible to change this structure without taking the data model into account. By way of example, a client is able to perform copy, move or rename operations using the IStorage interface without an OVA data model server being involved. This entails the problem that inconsistencies can arise which later need to be corrected again manually by the user.
This now poses the problem of how consistency can be produced as far as possible without demanding excessive effort from the developer of ES-Autos or OVA tools for this purpose. These mechanisms should also not be

transparent to the parts of the API which are used by the

OVA tools. A client application should always be occupied with the ES-Auto names, for example, and not with cryptic IDs (see FIGURE 1).

- ***Problematical actions***

It is first necessary to examine the actions which conceal potential risks. These are, firstly, all unidirectional relationships, and, secondly, actions which "pass by" the data model servers.

- Unidirectional links

Unidirectional links are problematical because it is not possible to tell from an action that an inconsistency is being produced. If a link is used to point to a file in a Word document, for example, and this file is renamed at a later point in time, the Word document is not told anything about this and will not find the file again. This problem can be eliminated only by a central authority which knows where the file can be found.

- "Dumb" actions

In this regard, dumb actions denote actions carried out without the knowledge of the data model. An example is renaming an object using the IStorage interface (IStorage::RenameElement). Such actions are always possible for standardized data access. In this case too, a central authority could help to minimize the problem. An important aspect in both cases is, above all, error recognition, and if possible also error elimination.

- ***Object ID moniker***

As described above, a central office can manage the objects such that they are (virtually) uniquely identifiable. All the objects are therefore referenced using object IDs which can be resolved by the central

office, in our case the Active Directory Service.
This ID is independent of all actions; it is allocated upon creation of the object and then does not change again

so long as no other object having the same ID exists. This will only occur in the case of copying outside the data model, however.

Each container allocates a name when an embedded object is created.

FIGURE 2 shows a schematic illustration of the allocation and assignment of object identifications as object IDs. These IDs, in the figure ID1, ID2, ID3 and ID4, are respectively stored with their container. This means that the hierarchy container knows ID1 and ID2, Chart 2 knows ID3. In this regard, these IDs are unique only within the container on the topmost level. This means that Chart 1 can also start with ID1 again. Individual objects can now be identified using a chain of IDs. ES-Auto 1 is identified using /ID1!ID1, for example. The connection between ES-Auto 2 and ES-Auto 3 (IDy) is given the following IDs:

IDy = /ID1!ID4!c -> /ID2!ID3!d

IDy is thus a type of alias for the connection between ES-Auto 2 and ES-Auto 3. This alias is stored with the lowest possible container in the hierarchy. In this case, this is the H-container 1, since both Chart 1 and Chart 2 are involved in the connection.
The connection IDx involves only the two ES-Autos 1 and 2; the information relating to how IDx is resolved can therefore be stored with Chart 1. This procedure of keeping the information as local as possible has the advantage that these references can then be resolved even if only a subcontext is opened. In such a subcontext, all the references, connections etc. which remain within the context are thus known. Any references to the outside (or from the outside) cannot be resolved.

- Context IDs vs local IDs

As can be seen in the example above, there are two different types of IDs. First the *local* IDs, such as ID1, ID2, ..., which are only ever known locally to the container. Secondly, there are the *context* IDs, which have their validity within the whole of the current context. Both IDs need to be unique in their environment. The local ones at container level, the context IDs on a context-wide basis.

- Resolution

An ID needs to be resolved, for example, when the object is to be activated. Thus, for example, ES-Auto 2 will check its connections during a consistency check. To this end, IDx and IDy need to be ascertained. To do this, ES-Auto 2 ascertains the individual objects from the context. Since the connections are stored as monikers, a BindToObject() is sufficient from the standpoint of the ES-Auto:

```
...
MkParseDisplayName ("@objectID!IDy!Source",&pMoniker);
pConnector = pMoniker->BindToObject ();
...
```

Since the monikers in this case are ObjectID monikers, the server for ObjectID monikers, that is to say the context, is asked for the object. The context knows IDy, because it is responsible for storing it, and resolves it into its component parts. The ParseDisplayName function extracts IDy and Source and establishes that this is /ID1!ID4!c. The containers are now recursively asked for this object.

This somewhat more complicated method is advantageous because the (possible) connections are also available in the subcontexts. In the example above, GR 99 P 3135

Chart 1 could also be chosen as the entry point (context). It is then also possible to access the connection IDx. In this case, IDy is an external connection which is not available so long as the operator is moving only in the context of Chart 1.

- ***Use/examples***

The effects are now shown using a few examples. The actions are move, copy, delete and rename.

- Move

The initial situation is **Error! Reference source could not be found.**. ES-Auto 1 is now moved from Chart 1 to Chart 2 (see **Error! Reference source could not be found.**). This is done in two different ways. Firstly in an OVA tool, secondly past the data model ("dumb moving").

Moving in the OVA tool

If the move operation is initiated in an OVA tool, the servers for the data model adopt the action and thus ensure that all references remain valid. The agitators involved in this case are the two charts. In the source chart, the method MoveESAuto is initiated. This method is provided with the ES-Auto and the destination chart:

```
...
pChart1->MoveESAuto ("ESAuto 1", pChart2);
...
```

The method MoveESAuto thus encapsulates all the necessary steps. These are, firstly, copying the ES-Auto into Chart 2, then deleting the original auto from Chart 1 and finally matching the IDs. Since, with such an action, only the IDs up to the object on which the action was carried out can ever be altered, only this need actually be notified to the context:

```
...
```

```
pContext->UpdateReference ("ID1!ID1", "ID2!ID4");
...
```

This method prompts the context to change all IDs which start with "ID1!ID1" to "ID2!ID4".

FIGURE 3 shows a schematic illustration for moving an object denoted by "ES-Auto1".

"Dumb" moving

With "dumb" copying, the servers for the object model are not involved. This means that the IDs cannot be updated either at this point in time. To perform such an action, it is sufficient to move the persistent data store for ES-Auto 1 from the store for Chart 1 to that for Chart 2. When Chart 1 is opened, it will no longer display ES-Auto 1 and will delete its entry for "ID1". When Chart 2 is opened, it will establish that an ES-Auto for which no ID has been allocated yet has been added to its data store. ES-Auto 1 is therefore now assigned an ID. In addition, the references for ES-Auto 1 and the partners involved still need to be replaced. In the case of bidirectional IDs, this is not a problem. If ES-Auto 1 is asked for its external references, it will return IDx. If the context is asked for this ID, it can only resolve part ("ID1!ID4") immediately. "ID1!ID1" still points to nothing at this time. Since the action has been triggered by checking ES-Auto 1, however, this can be remedied (possibly on the basis of a query to the user):

```
...
Foreach id In pESAuto1->GetExternalRefs ()
   bOK = CheckReference (id.Source);
   if ! bOK
     UpdateReference (id.Source, "ID2!ID4");
   Endif

   bOK = CheckReference (id.Destination);
   if ! bOK
     UpdateReference (id.Destination, "ID2!ID4");
   Endif
Endfor
```

- Copy

The two methods will also be examined in the case of copying:

Copying in the OVA tool

If ES-Auto 1 is only copied, nothing needs to be changed on the references. For the destination ES-Auto, Chart 2 needs to allocate a new ID (for example ID4). In addition, all the external references for the new ES-Auto should be deleted.

"Dumb" copying

If ES-Auto 1 is again copied past the object model, as in the case of moving, 2 ES-Autos refer to ES-Auto 2. This is not a problem, however, since in this case too, Chart 2 establishes that ES-Auto 1 is still not known to it (no ID). As in the case of moving, it will now allocate a new ID and test the external references of the ES-Auto. In this case, ES-Auto 1 also exists in Chart 1, however. This check will therefore return no error. It is therefore necessary to check whether the unknown ES-Auto is part of the external reference. The code from above therefore needs to be extended a little further:

```
...
Foreach id In pESAuto1->GetExternalRefs ()
   bOK = CheckReference (id.Source);
   If ! bOK
     UpdateReference (id.Source, "ID2!ID4");
   else
     If ! IsReferenceParticipant ("ID2!ID4")
          pESAuto1->RemoveReference("ID2!ID4");
     Endif
   Endif

   bOK = CheckReference (id.Destination);
   if ! bOK
     UpdateReference (id.Destination, "ID2!ID4");
   else
     If ! IsReferenceParticipant ("ID2!ID4")
          pESAuto1->RemoveReference("ID2!ID4");
     Endif
   Endif
Endfor
```

- Delete

When deleting an ES-Auto, the following steps arise:

Deletion in the OVA tool

If the ES-Auto is deleted, all the references can also be deleted immediately. In this case, the context is notified that a particular ID is no longer valid. This can now forward to all (on the basis of user query) objects having a reference to this ID the message that the ID is invalid. If ES-Auto 1 is deleted in **Error! Reference source could not be found.**, for example, the following code needs to be executed:

```
...
RemoveReference ("ID1!ID1");
...
```

The result of this call is that the context calls the method RemoveExternalReference () for all partners.

"Dumb" deletion

With dumb deletion, two situations can arise: the first possibility is for Chart 1 to be opened first. This chart establishes that an ES-Auto no longer exists for the ID1. It now deletes its entry "ID1".
On the other hand, an object can attempt to resolve the reference (e.g. ES-Auto 2). Since this is not possible, the user is asked what has happened to ES-Auto 1, and what is meant to happen to the reference.

- Rename

Renaming is similar to moving.

In summary, the invention relates to a system and method for object identification in distributed hierarchical systems, in particular in automation systems. To ensure object identification for operations such as moving, copying, renaming, etc., the invention proposes introducing contexts for forming a plurality of indirection stages for managing identifiers. This provides efficient methods for repairing "broken links" without introducing global, central management functions.

Patent Claims

1. A system for object identification in distributed hierarchical systems, in particular in automation systems having means for forming a plurality of indirection stages for managing identifiers for objects.

2. A method for object identification in distributed hierarchical systems, in particular in automation systems, in which the objects are identified by a plurality of indirection stages for managing identifiers.

09/936203
518 Rec'd PCT/PTO 10 SEP 2001

Substitute Specification

TITLE OF THE INVENTION

SYSTEM AND METHOD FOR IDENTIFYING OBJECTS IN DISTRIBUTED HIERARCHICAL SYSTEMS, IN PARTICULAR IN AUTOMATION SYSTEMS

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The invention relates to a system and method for object identification in distributed hierarchical systems, in particular in automation systems. Such a system and method are used particularly in the field of automation technology.

[0002]

2. Description of the Related Art

[0003] The invention is based on the insight that previous solutions are not very robust and/or require a great deal of modification effort. There are two basic identification mechanisms which are used (and can also be combined with one another). One method is based on object identification by allocating a globally unique identifier for each object. This global identifier is used to ensure that an object can be found again irrespective of its present whereabouts. This method has the following drawbacks:

- **Central management**: The method requires central management structures such as management of the object identifiers and conversion tables for the object identifiers to the objects.
- **Poor support for distributed work**: The need for central management complicates the splitting of object sets, separate processing and subsequent merging thereof (headword: branch and merge).

[0004] In the second method, an object is identified by its relative position with respect to another. This then also stipulates how the object can be found. In contrast to the first method, an object does not have a unique identifier, but instead the identifier dependent on the respective starting object referencing the other one. This means that no central management information is required. However, the following drawbacks result:

- **Low robustness**: Using the relative position for identification means that the identifier (or the identifiers) becomes invalid when the object is moved, and the object is no longer available (broken link).
- **Great deal of modification effort**: When the identifiers for an object have become invalid, they need to be corrected using a type of correction operation.

SUMMARY OF THE INVENTION

[0005] An object of the invention is to ensure object identification for operations such as moving, copying, renaming, etc.

[0006] With the inventive solution, contexts are introduced for forming a plurality of indirection stages for managing the identifiers. This provides efficient methods for repairing "broken links" without introducing global, central management functions.

- **No central management**: Management is through the context hierarchy. This means that each context contains all the necessary information.
- **Support for distributed work**: The context hierarchy can be broken down and reassembled as desired. This means that projects can be branched and merged without difficulty.
- **Little modification effort**: The context hierarchy makes it immediately clear where changes to identifiers are to be reconstructed. The changes also need to be made only on the context objects in question.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The invention is described and explained in more detail below using the exemplary embodiments shown in the figures, in which:

FIGURE 1 is a block diagram of a method to identify the facts of the matter: a client sees names, an object model works with IDs,

FIGURE 2 is a schematic object diagram for the allocation and assignment of object identifications as object IDs, and

FIGURE 3 is a schematic object diagram for the moving of an object denoted by "ES-Auto1".

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0008] The method is described below within the context of the OVA engineering object model (OVA= open distributed automation). It can also be used for other object models, however. For a better understanding of the relationships, a brief description of the context of the invention will be given below:

[0009] For each object, there is an environment in which it is known. In the case of OVA, this environment is modeled by the context. Within a context, names and identifiers for all contained objects are known and unique. The context is generally determined by the point of entry chosen by a user for processing his automation solution. Context information is available on any container object (i.e. an object containing other objects), such as H-container, chart or master. The smallest environment for a context is a document, however. For the case of embedded objects, the context information of the surrounding document is used. However, this also means that context information can be structured hierarchically. In this case, contexts at a lower point are always part of the higher context in the hierarchy as well. In addition, further contexts which are not hierarchically related can be associated with any desired context. This context can then access the information of the associated context. This association is unidirectional, however. In the case of associated contexts, the (hierarchically) contained contexts are also associated automatically.

[0010] The context information determines which objects have been entered in the directory. The content of the document automatically belongs to the same context; this also applies, in particular, to linked objects or objects which have been added using a rule ("all objects in this directory", etc.). The context is also the manager of the context IDs. These are described later.

Object identification

[0011] Since OVA uses a standardized method to access the data storage, it is necessary to structure the objects so that they can be safely identified. The reason for this is that some of the data are structure information, for example which ES-Auto is located in which chart, or which auto is connected to which. This structure information is subject to rules which are taken into account by the implementation of the data model. In the case of the current realization using the IStorage interface as the interface for data storage, it is always possible to change this structure without taking the data model into account. By way of example, a client is able to perform copy, move or rename operations using the IStorage interface without an OVA data model server being involved. This entails the problem that inconsistencies can arise which later need to be corrected again manually by the user.

[0012] This now poses the problem of how consistency can be produced as far as possible without demanding excessive effort from the developer of ES-Autos or OVA tools for this purpose. These mechanisms should also not be transparent to the parts of the API which are used by the OVA tools. A client application should always be occupied with the ES-Auto names, for example, and not with cryptic IDs (see FIGURE 1).

- ***Problematical actions***

It is first necessary to examine the actions which conceal potential risks. These are, firstly, all unidirectional relationships, and, secondly, actions which "pass by" the data model servers.

- Unidirectional links

Unidirectional links are problematical because it is not possible to tell from an action that an inconsistency is being produced. If a link is used to point to a file in a Word document, for example, and this file is renamed at a later point in time, the Word document is not told anything about this and will not find the file again. This problem can be eliminated only by a central authority which knows where the file can be found.

- "Dumb" actions

In this regard, dumb actions denote actions carried out without the knowledge of the data model. An example is renaming an object using the IStorage interface (IStorage::RenameElement). Such actions are always possible for standardized data access. In this case too, a central authority could help to minimize the problem. An important aspect in both cases is, above all, error recognition, and if possible also error elimination.

- ***Object ID moniker***

As described above, a central office can manage the objects such that they are (virtually) uniquely identifiable. All the objects are therefore referenced using object IDs which can be resolved by the central office, in our case the Active Directory Service. This ID is independent of all actions; it is allocated upon creation of the object and then does not change again so long as no other object having the same ID exists. This will only occur in the case of copying outside the data model, however. Each container allocates a name when an embedded object is created.

[0013] FIGURE 2 is a schematic illustration of the allocation and assignment of object identifications as object IDs. These IDs, in the figure ID1, ID2, ID3 and ID4, are respectively stored with their container. This means that the hierarchy container knows ID1 and ID2, Chart 2 knows ID3. In this regard, these IDs are unique only within the container on the topmost level. This means that Chart 1 can also start with ID1 again. Individual objects can now be identified using a chain of IDs. ES-Auto 1 is identified using /ID1!ID1, for example. The connection between ES-Auto 2 and ES-Auto 3 (IDy) is given the following IDs:

IDy = /ID1!ID4!c -> /ID2!ID3!d

[0014] IDy is thus a type of alias for the connection between ES-Auto 2 and ES-Auto 3. This alias is stored with the lowest possible container in the hierarchy. In this case, this is the H-container 1, since both Chart 1 and Chart 2 are involved in the connection. The connection IDx involves only the two ES-Autos 1 and 2; the information relating to how IDx is resolved can therefore be stored with Chart 1. This procedure of keeping the information as local as possible has the advantage that these references can then be resolved even if only a subcontext is opened. In such a subcontext, all the references, connections etc. which remain within the context are thus known. Any references to the outside (or from the outside) cannot be resolved.

- Context IDs vs local IDs

[0015] As can be seen in the example above, there are two different types of IDs. First the *local* IDs, such as ID1, ID2, ..., which are only ever known locally to the container. Secondly, there are the *context* IDs, which have their validity within the whole of the current context. Both IDs need to be unique in their environment. The local ones at container level, the context IDs on a context-wide basis.

- Resolution

[0016] An ID needs to be resolved, for example, when the object is to be activated. Thus, for example, ES-Auto 2 will check its connections during a consistency check. To this end, IDx and IDy need to be ascertained. To do this, ES-Auto 2 ascertains the individual objects from the context. Since the connections are stored as monikers, a BindToObject() is sufficient from the standpoint of the ES-Auto:

```
...
MkParseDisplayName ("@objectID!IDy!Source",&pMoniker);
pConnector = pMoniker->BindToObject ();
...
```

[0017] Since the monikers in this case are ObjectID monikers, the server for ObjectID monikers, that is to say the context, is asked for the object. The context knows IDy, because it is responsible for storing it, and resolves it into its component parts. The ParseDisplayName function extracts IDy and Source and establishes that this is /ID1!ID4!c. The containers are now recursively asked for this object.

[0018] This somewhat more complicated method is advantageous because the (possible) connections are also available in the subcontexts. In the example above, Chart 1 could also be chosen as the entry point (context). It is then also possible to access the connection IDx. In this case, IDy is an external connection which is not available so long as the operator is moving only in the context of Chart 1.

- ***Use/examples***

[0019] The effects are now shown using a few examples. The actions are move, copy, delete and rename.

- Move

[0020] The initial situation is **Error! Reference source could not be found.**. ES-Auto 1 is now moved from Chart 1 to Chart 2 (see **Error! Reference source could not be found.**). This is done in two different ways. Firstly in an OVA tool, secondly past the data model ("dumb moving").

[0021] Moving in the OVA tool

[0022] If the move operation is initiated in an OVA tool, the servers for the data model adopt the action and thus ensure that all references remain valid. The agitators involved in this case are the two charts. In the source chart, the method MoveESAuto is initiated. This method is provided with the ES-Auto and the destination chart:

```
...
pChart1->MoveESAuto ("ESAuto 1", pChart2);
...
```

[0023] The method MoveESAuto thus encapsulates all the necessary steps. These are, firstly, copying the ES-Auto into Chart 2, then deleting the original auto from Chart 1 and finally matching the IDs. Since, with such an action, only the IDs up to the object on which the action was carried out can ever be altered, only this need actually be notified to the context:

```
...
pContext->UpdateReference ("ID1!ID1", "ID2!ID4");
...
```

This method prompts the context to change all IDs which start with "ID1!ID1" to "ID2!ID4".

[0024] FIGURE 3 shows a schematic illustration for moving an object denoted by "ES-Auto1".

"Dumb" moving

[0025] With "dumb" copying, the servers for the object model are not involved. This means that the IDs cannot be updated either at this point in time. To perform such an action, it is sufficient to move the persistent data store for ES-Auto 1 from the store for Chart 1 to that for Chart 2. When Chart 1 is opened, it will no longer display ES-Auto 1 and will delete its entry for "ID1". When Chart 2 is opened, it will establish that an ES-Auto for which no ID has been allocated yet has been added to its data store. ES-Auto 1 is therefore now assigned an ID. In addition, the references for ES-Auto 1 and the partners involved still need to be replaced. In the case of bidirectional IDs, this is not a problem. If ES-Auto 1 is asked for its external references, it will return IDx. If the context is asked for this ID, it can only resolve part ("ID1!ID4") immediately. "ID1!ID1" still points to nothing at this time. Since the action has been triggered by checking ES-Auto 1, however, this can be remedied (possibly on the basis of a query to the user):

```
...
Foreach id In pESAuto1->GetExternalRefs ()
    bOK = CheckReference (id.Source);
    if ! bOK
        UpdateReference (id.Source, "ID2!ID4");
    Endif
```

```
        bOK = CheckReference (id.Destination);
        if ! bOK
            UpdateReference (id.Destination, "ID2!ID4");
        Endif
    Endfor
```

[0026]

- Copy

[0027] The two methods will also be examined in the case of copying:

Copying in the OVA tool

[0028] If ES-Auto 1 is only copied, nothing needs to be changed on the references. For the destination ES-Auto, Chart 2 needs to allocate a new ID (for example ID4). In addition, all the external references for the new ES-Auto should be deleted.

"Dumb" copying

[0029] If ES-Auto 1 is again copied past the object model, as in the case of moving, 2 ES-Autos refer to ES-Auto 2. This is not a problem, however, since in this case too, Chart 2 establishes that ES-Auto 1 is still not known to it (no ID). As in the case of moving, it will now allocate a new ID and test the external references of the ES-Auto. In this case, ES-Auto 1 also exists in Chart 1, however. This check will therefore return no error. It is therefore necessary to check whether the unknown ES-Auto is part of the external reference. The code from above therefore needs to be extended a little further:

```
...
Foreach id In pESAuto1->GetExternalRefs ()
    bOK = CheckReference (id.Source);
    If ! bOK
        UpdateReference (id.Source, "ID2!ID4");
    else
        If ! IsReferenceParticipant ("ID2!ID4")
            pESAuto1->RemoveReference("ID2!ID4");
        Endif
    Endif
```

```
        bOK = CheckReference (id.Destination);
        if ! bOK
            UpdateReference (id.Destination, "ID2!ID4");
        else
            If ! IsReferenceParticipant ("ID2!ID4")
                pESAuto1->RemoveReference("ID2!ID4");
            Endif
        Endif
    Endfor
```

- Delete

[0030] When deleting an ES-Auto, the following steps arise:

Deletion in the OVA tool

[0031] If the ES-Auto is deleted, all the references can also be deleted immediately. In this case, the context is notified that a particular ID is no longer valid. This can now forward to all (on the basis of user query) objects having a reference to this ID the message that the ID is invalid. If ES-Auto 1 is deleted in **Error! Reference source could not be found.**, for example, the following code needs to be executed:

```
...
RemoveReference ("ID1!ID1");
...
```

The result of this call is that the context calls the method RemoveExternalReference () for all partners.

"Dumb" deletion

[0032] With dumb deletion, two situations can arise: the first possibility is for Chart 1 to be opened first. This chart establishes that an ES-Auto no longer exists for the ID1. It now deletes its entry "ID1".

[0033] On the other hand, an object can attempt to resolve the reference (e.g. ES-Auto 2). Since this is not possible, the user is asked what has happened to ES-Auto 1, and what is meant to happen to the reference.

- Rename

[0034] Renaming is similar to moving.

[0035] In summary, the invention relates to a system and method for object identification in distributed hierarchical systems, in particular in automation systems. To ensure object identification for operations such as moving, copying, renaming, etc., the invention proposes introducing contexts for forming a plurality of indirection stages for managing identifiers. This provides efficient methods for repairing "broken links" without introducing global, central management functions.

Substitute Abstract

ABSTRACT OF DISCLOSURE

SYSTEM AND METHOD FOR OBJECT IDENTIFICATION IN DISTRIBUTED HIERARCHICAL SYSTEMS, IN PARTICULAR IN AUTOMATION SYSTEMS

The invention relates to a system and method for object identification in distributed hierarchical systems, in particular in automation systems. To ensure object identification for operations such as moving, copying, renaming, etc., the invention proposes introducing contexts for forming a plurality of indirection stages for managing identifiers. This provides efficient methods for repairing “broken links” without introducing global, central management functions.

09/936203

GR 99 P 3135

1/2

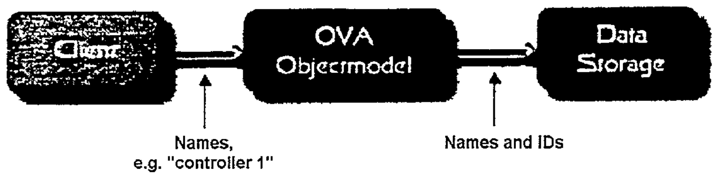

Fig. 1

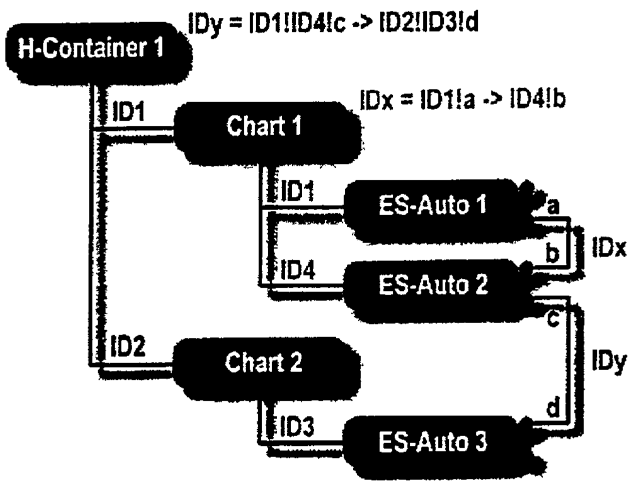

Fig. 2

09/936203

GR 99 P 3135

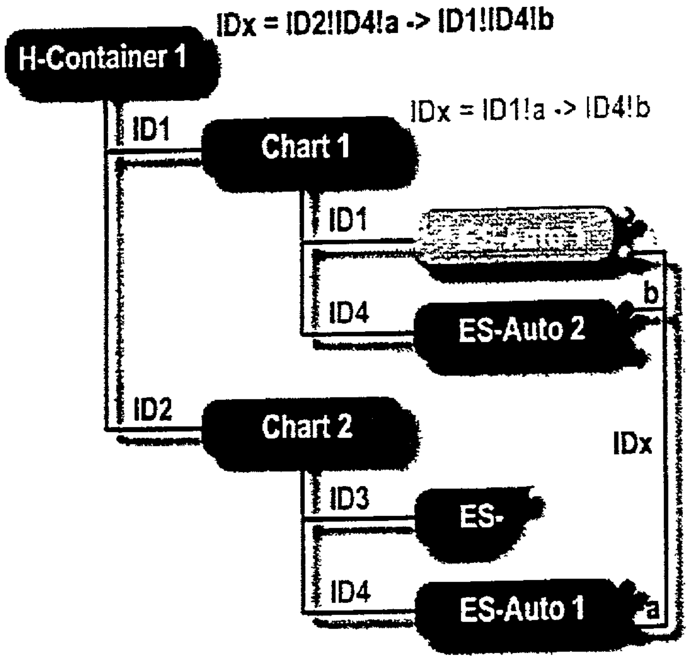

Fig. 3

Declaration and Power of Attorney For Patent Application
Erklärung Für Patentanmeldungen Mit Vollmacht
German Language Declaration

Als nachstehend benannter Erfinder erkläre ich hiermit an Eides Statt:

dass mein Wohnsitz, meine Postanschrift, und meine Staatsangehörigkeit den im Nachstehenden nach meinem Namen aufgeführten Angaben entsprechen,

dass ich, nach bestem Wissen der ursprüngliche, erste und alleinige Erfinder (falls nachstehend nur ein Name angegeben ist) oder ein ursprünglicher, erster und Miterfinder (falls nachstehend mehrere Namen aufgeführt sind) des Gegenstandes bin, für den dieser Antrag gestellt wird und für den ein Patent beantragt wird für die Erfindung mit dem Titel:

System und Verfahren zur Objektidentifizierung in verteilten hierarchischen Systemen, insbesondere in Automatisierungssystemen

deren Beschreibung

(zutreffendes ankreuzen)
☐ hier beigefügt ist.
☒ am 09.03.2000 als
PCT internationale Anmeldung
PCT Anmeldungsnummer PCT/DE00/00738
eingereicht wurde und am ______
abgeändert wurde (falls tatsächlich abgeändert).

Ich bestätige hiermit, dass ich den Inhalt der obigen Patentanmeldung einschliesslich der Ansprüche durchgesehen und verstanden habe, die eventuell durch einen Zusatzantrag wie oben erwähnt abgeändert wurde.

Ich erkenne meine Pflicht zur Offenbarung irgendwelcher Informationen, die für die Prüfung der vorliegenden Anmeldung in Einklang mit Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) von Wichtigkeit sind, an.

Ich beanspruche hiermit ausländische Prioritätsvorteile gemäss Abschnitt 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 119 aller unten angegebenen Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde, und habe auch alle Auslandsanmeldungen für ein Patent oder eine Erfindersurkunde nachstehend gekennzeichnet, die ein Anmeldedatum haben, das vor dem Anmeldedatum der Anmeldung liegt, für die Priorität beansprucht wird.

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

System and method for identifying objects in distributed hierarchical systems, especially automation systems

the specification of which

(check one)
☐ is attached hereto.
☒ was filed on 09.03.2000 as
PCT international application
PCT Application No. PCT/DE00/00738
and was amended on ______
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

German Language Declaration

Prior foreign appplications
Priorität beansprucht

			Priority Claimed	
19910527.8 (Number) (Nummer)	DE (Country) (Land)	09.03.1999 (Day Month Year Filed) (Tag Monat Jahr eingereicht)	☒ Yes Ja	☐ No Nein
(Number) (Nummer)	(Country) (Land)	(Day Month Year Filed) (Tag Monat Jahr eingereicht)	☐ Yes Ja	☐ No Nein
(Number) (Nummer)	(Country) (Land)	(Day Month Year Filed) (Tag Monat Jahr eingereicht)	☐ Yes Ja	☐ No Nein

Ich beanspruche hiermit gemäss Absatz 35 der Zivilprozessordnung der Vereinigten Staaten, Paragraph 120, den Vorzug aller unten aufgeführten Anmeldungen und falls der Gegenstand aus jedem Anspruch dieser Anmeldung nicht in einer früheren amerikanischen Patentanmeldung laut dem ersten Paragraphen des Absatzes 35 der Zivilprozeßordnung der Vereinigten Staaten, Paragraph 122 offenbart ist, erkenne ich gemäss Absatz 37, Bundesgesetzbuch, Paragraph 1.56(a) meine Pflicht zur Offenbarung von Informationen an, die zwischen dem Anmeldedatum der früheren Anmeldung und dem nationalen oder PCT internationalen Anmeldedatum dieser Anmeldung bekannt geworden sind.

I hereby claim the benefit under Title 35. United States Code. §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §122, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occured between the filing date of the prior application and the national or PCT international filing date of this application.

PCT/DE00/00738 (Application Serial No.) (Anmeldeseriennummer)	09.03.2000 (Filing Date D, M, Y) (Anmeldedatum T, M, J)	(Status) (patentiert, anhängig, aufgegeben)	pending (Status) (patented, pending, abandoned)
(Application Serial No.) (Anmeldeseriennummer)	(Filing Date D,M,Y) (Anmeldedatum T, M; J)	(Status) (patentiert, anhängig, aufgeben)	(Status) (patented, pending, abandoned)

Ich erkläre hiermit, dass alle von mir in der vorliegenden Erklärung gemachten Angaben nach meinem besten Wissen und Gewissen der vollen Wahrheit entsprechen, und dass ich diese eidesstattliche Erklärung in Kenntnis dessen abgebe, dass wissentlich und vorsätzlich falsche Angaben gemäss Paragraph 1001, Absatz 18 der Zivilprozessordnung der Vereinigten Staaten von Amerika mit Geldstrafe belegt und/oder Gefängnis bestraft werden koennen, und dass derartig wissentlich und vorsätzlich falsche Angaben die Gültigkeit der vorliegenden Patentanmeldung oder eines darauf erteilten Patentes gefährden können.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

German Language Declaration

VERTRETUNGSVOLLMACHT: Als benannter Erfinder beauftrage ich hiermit den nachstehend benannten Patentanwalt (oder die nachstehend benannten Patentanwälte) und/oder Patent-Agenten mit der Verfolgung der vorliegenden Patentanmeldung sowie mit der Abwicklung aller damit verbundenen Geschäfte vor dem Patent- und Warenzeichenamt: *(Name und Registrationsnummer anführen)*	POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

And I hereby appoint

Customer No. 21171

Telefongespräche bitte richten an: *(Name und Telefonnummer)*	Direct Telephone Calls to: *(name and telephone number)* Ext. ________

Postanschrift: Send Correspondence to:

Staas & Halsey LLP
700 Eleventh Street NW, Suite 500 20001 Washington, DC
Telephone: (001) 202 434 1500 and Facsimile (001) 202 434 1501
or
Customer No. 21171

Voller Name des einzigen oder ursprünglichen Erfinders: NORBERT BECKER	Full name of sole or first inventor: NORBERT BECKER
Unterschrift des Erfinders Datum	Inventor's signature Norbert Becker Date 22.8.2001
Wohnsitz ERLANGEN, DEUTSCHLAND	Residence ERLANGEN, GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift TURMHÜGELWEG 20A	Post Office Addess TURMHÜGELWEG 20A
91058 ERLANGEN DEUTSCHLAND	91058 ERLANGEN GERMANY
Voller Name des zweiten Miterfinders (falls zutreffend): GEORG BIEHLER	Full name of second joint inventor, if any: GEORG BIEHLER
Unterschrift des Erfinders Datum	Second Inventor's signature [illegible] Date 22.8.2001
Wohnsitz NÜRNBERG, DEUTSCHLAND	Residence NÜRNBERG, GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift SCHALKHAUSSER STR. 102A	Post Office Address SCHALKHAUSSER STR. 102A
90473 NÜRNBERG DEUTSCHLAND	90473 NÜRNBERG GERMANY

100

200

10030711 . 022502

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben). *(Supply similar information and signature for third and subsequent joint inventors).*

Voller Name des dritten Miterfinders: MATTHIAS DIEZEL 300	Full name of third joint inventor: MATTHIAS DIEZEL
Unterschrift des Erfinders Datum	Inventor's signature [illegible] Date 10.9.01
Wohnsitz LAUFAMHOLZ, DEUTSCHLAND	Residence ~~LAUFAMHOLZ,~~ GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift GLÄSLEINSACKERWEG 25	Post Office Address GLÄSLEINSACKERWEG 25
90482 LAUFAMHOLZ DEUTSCHLAND	90482 LAUFAMHOLZ GERMANY
Voller Name des vierten Miterfinders: Dr. ALBRECHT DONNER 400	Full name of fourth joint inventor: Dr. ALBRECHT DONNER
Unterschrift des Erfinders Datum	Inventor's signature [illegible] Date 20.9.01
Wohnsitz MARKERSDORF, DEUTSCHLAND	Residence ~~MARKERSDORF~~, GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift HAUPTSTR.92	Post Office Address HAUPTSTR.92
09236 MARKERSDORF DEUTSCHLAND	09236 MARKERSDORF GERMANY
Voller Name des fünften Miterfinders: Dr. DIETER ECKARDT 500	Full name of fifth joint inventor: Dr. DIETER ECKARDT
Unterschrift des Erfinders Datum	Inventor's signature Dieter Eckardt Date 18.09.2001
Wohnsitz HERZOGENAURACH, DEUTSCHLAND	Residence ~~HERZOGENAURACH,~~ GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift ZIEHRER STR 8	Post Office Address ZIEHRER STR 8
91074 HERZOGENAURACH DEUTSCHLAND	91074 HERZOGENAURACH GERMANY
Voller Name des sechsten Miterfinders: MANFRED KRÄMER 600	Full name of sixth joint inventor. MANFRED KRÄMER
Unterschrift des Erfinders Datum	Inventor's signature [illegible] Date 02.10.01
Wohnsitz WENDELSTEIN, DEUTSCHLAND	Residence ~~WENDELSTEIN,~~ GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift FLIEDERWEG 21A	Post Office Address FLIEDERWEG 21A
90530 WENDELSTEIN DEUTSCHLAND	90530 WENDELSTEIN GERMANY

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).

(Supply similar information and signature for third and subsequent joint inventors).

10030711 . 032560

7-00

Voller Name des siebten Miterfinders: DIRK LANGKAFEL	Full name of seventh joint inventor: DIRK LANGKAFEL
Unterschrift des Erfinders Datum	Inventor's signature [illegible] Date 12.09.01
Wohnsitz EFFELTRICH, DEUTSCHLAND	Residence EFFELTRICH, GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift BERGSTR. 15A	Post Office Address BERGSTR. 15A
91090 EFFELTRICH DEUTSCHLAND	91090 EFFELTRICH GERMANY

8-00

Voller Name des achten Miterfinders (falls zutreffend): RALF LEINS	Full name of eighth joint inventor, if any: RALF LEINS
Unterschrift des Erfinders Datum	Inventor's signature [illegible] Date 28.8.01
Wohnsitz ISPRINGEN, DEUTSCHLAND	Residence ISPRINGEN, GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift IM MAHLER 38	Post Office Address IM MAHLER 38
75228 ISPRINGEN DEUTSCHLAND	75228 ISPRINGEN GERMANY

9-00

Voller Name des neunten Miterfinders (falls zutreffend). RONALD LANGE	Full name of nineth joint inventor, if any: RONALD LANGE
Unterschrift des Erfinders Datum	Inventor's signature [illegible] Date 8/23/01
Wohnsitz FÜRTH, DEUTSCHLAND	Residence FÜRTH, GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift VIRCHOWSTR. 28	Post Office Address VIRCHOWSTR. 28
90766 FÜRTH DEUTSCHLAND	90766 FÜRTH GERMANY

10-00

Voller Name des zehnten Miterfinders (falls zutreffend). KARSTEN SCHNEIDER	Full name of tenth joint inventor, if any. KARSTEN SCHNEIDER
Unterschrift des Erfinders Datum	Inventor's signature [illegible] Date 11.09.01
Wohnsitz ERLANGEN, DEUTSCHLAND	Residence ERLANGEN, GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift BOHLENPLATZ 7	Post Office Address BOHLENPLATZ 7
91054 ERLANGEN DEUTSCHLAND	91054 ERLANGEN GERMANY

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).

(Supply similar information and signature for third and subsequent joint inventors).

11-00

09936203.110801

Voller Name des elften Miterfinders: HELMUT WINDL	Full name of eleventh joint inventor: HELMUT WINDL
Unterschrift des Erfinders Datum	Inventor's signature [illegible] Date 8/29/2001
Wohnsitz PEISIG, DEUTSCHLAND	Residence PEISIG, GERMANY DEX
Staatsangehörigkeit DEUTSCH	Citizenship GERMAN
Postanschrift FÖHRENSTR.10	Post Office Address FÖHRENSTR.10
93077 PEISIG DEUTSCHLAND	93077 PEISIG GERMANY

Voller Name des zwölften Miterfinders (falls zutreffend)	Full name of twelvth joint inventor, if any:
Unterschrift des Erfinders Datum	Inventor's signature Date
Wohnsitz ,	Residence ,
Staatsangehorigkeit	Citizenship
Postanschrift	Post Office Address

Voller Name des dreizehnten Miterfinders (falls zutreffend):	Full name of thirteenth joint inventor, if any:
Unterschrift des Erfinders Datum	Inventor's signature Date
Wohnsitz ,	Residence ,
Staatsangehörigkeit	Citizenship
Postanschrift	Post Office Address

Voller Name des vierzehnten Miterfinders (falls zutreffend):	Full name of fourteenth joint inventor, if any:
Unterschrift des Erfinders Datum	Inventor's signature Date
Wohnsitz ,	Residence ,
Staatsangehörigkeit	Citizenship
Postanschrift	Post Office Address

(Bitte entsprechende Informationen und Unterschriften im Falle von dritten und weiteren Miterfindern angeben).

(Supply similar information and signature for third and subsequent joint inventors).